

CTIS 210 Final Project: Hunting With Rakko

My final project is a gacha-inspired game that incorporates clicker game and turn-based fighting game aspects. When the player starts the game, they are greeted with a home page containing: a “cake” clicker, a tutorial button for the entire game, buttons to increase and decrease health and attack, a button to gamble for more cake, and a panel at the bottom that allows the player to enter combat. Cake is the name of the currency in the game, which is used to increase stats and to gamble for more cake. Cake can be earned through the clicker game, gambling, and winning against bosses.

The stats panel on the left employs a 1-to-1 conversion of cake to attack or cake to health. Once the player increases attack, the attack will not change unless the player chooses to increase/decrease it. However, during battle, health can be decreased by the boss, and whether or not the battle is won, the health will remain decreased. The stats can be increased one at a time or ten at a time, but the game requires the player to have the corresponding number of cakes to change stats.

The gambling panel allows the player to spend 50 cakes for a chance to receive up to 500 cakes. However, as with most gambling games, the chances are not in the player’s favor. The percentages and possible results are: 1% for 500 cakes, 3% for 200 cakes, 10% for 100 cakes, 25% for 75 cakes, 25% for 25 cakes, and 36% for nothing. In my experience testing the game, I didn’t find it worth it to gamble, but it’s a fun option for those who enjoy it.

At the bottom, the first boss is automatically unlocked, but the other two are locked until the previous one is defeated. The player can fight the bosses as many times as they want, and they can fight a boss again even if they have defeated it already.

Throughout the home page, I tried my best to incorporate tutorials, information, and tips to make the game as easy to understand as possible. This way, players don't feel lost and overwhelmed when first opening the game.

In the boss fights, each one has the same mechanics, but with different amounts of attack and health. The first boss has 100 health and 20 attack, the second boss has 500 health and 50 attack, and the last boss has 1250 health and 150 attack. When the player starts the battle, they have a 50% chance of going first, and a 50% that the boss will go first. Each attack reduces the opponents health by that number of attack.

This game incorporates many different classes that we have gone over throughout the semester and many more that we did not discuss. The hardest part of creating this project was working with the layout managers and connecting the frames/panels together. For the layout managers, I used BorderLayout and BoxLayout to organize GUI components. The layout managers were rather finicky and did not put things where I wanted them to go, and it took a lot of panels within panels within panels to finally put things in the correct place. What was especially frustrating was trying to resize and align components, as the layout managers often restrict resizing. I've had experience with interfaces before using HTML/CSS/JavaScript, which I found much easier, but the layout managers did what I wanted to in the end.

I also struggled to connect the panels, frames, and variables together. We went over how to open multiple JFrames and connect signals from one JPanel to another, but I forgot that I needed to connect everything together in the main class and faced a lot of frustration and errors about things being null. However, once I figured out how to connect everything, the process became much easier.

Another challenge I faced was to load in a custom font. To be truly transparent, I ended up asking the embedded AI in VS Code to try and debug what was wrong, as I searched up how to do it but it was not working. I had the try-catch block, but I still could not figure out what was wrong with my code. I might have been able to figure it out on my own if I spent a lot more time, but I wanted to move on and work on other things.

I also could not figure out how to resize the ImageIcons within the code, so I ended up resizing them outside of code and adding the new images in to replace old ones. I think that I would need a little more knowledge and understanding of Java and layout managers to be able to successfully fix this problem of resizing within code, but I think given more time, I could figure it out.

As it is a game, there were some struggles to figure out how to adjust the difficulty of the game to ensure that it isn't too easy but also isn't too difficult. I think, through trial and error, I was able to create stats of the bosses that are relatively reasonable, especially considering the rewards for beating a boss and the ability to fight a boss as many times as you want and receiving the same rewards each time.

Since the game is inspired by a Japanese children's animation where all the characters are animals, I don't think I face the issue of character representation. Rakko is an already established character, known for their combat and hunting prowess, which I felt made them a good protagonist for this fighting game. Additionally, while I originally wanted to add a story to the game if I had the time, I was not able to include any narrative elements.

While I consider the game complete at this stage, there are some other features that I want to (and might actually implement on my own time) add on to the game. For one, I want to add an initial start page instead of immediately putting the player into the game. Another feature

I want to add is the critical hit system I detailed in a previous document. I also was not able to implement proportionally resizing items as the window changes size, which I think is important, especially considering limitations of different devices. I think I could also make parts of the code more efficient and consolidate certain parts under a method or a class.

Still, despite these challenges and future plans, I am satisfied with where the game is now and believe it's fun and easy to play.